Data Exploration and Visualization

McKell Stauffer and Maria Fabiano

December 14, 2017

1 Introduction

As technology rapidly advances, speech recognition is becoming more important in a myriad of applications (e.g. fighter aircrafts, disability assistance, and language learning)[1]. To enhance current speech recognition research, we analyze the fundamental statistics associated with a voice sample. More specifically, we focus a portion of this analysis on gender recognition. Improved capabilities in gender recognition from voice samples could have many beneficial implications for speech recognition systems [2].

Past research in gender recognition from a speech sample has analyzed fundamental frequencies, the slope of the spectra, mel-frequency cepstral coefficient, formant frequencies, and amplitudes[3]. From past research, the most basic fact of note is that the fundamental frequency of adult female voices are higher than adult male voices. Studies have found that the fundamental frequency of an adult male ranges from 85 to 180 Hz, and that of an adult female ranges from 165 to 255 Hz[4]. As there is an overlap in the range of fundamental frequencies, we are interested in finding other statistics to distinguish between the genders. More specifically, we are interested in studying the following two questions:

- Are there differences between the statistics associated with male and female voice samples?
- How do the various acoustic statistics relate to one another?

```
In [1]: import pandas as pd
        from collections import Counter
        import matplotlib.pyplot as plt
        import numpy as np
        %matplotlib inline
        plt.style.use("seaborn")
        plt.rcParams['figure.figsize'] = [10,6]
        plt.rcParams['figure.dpi'] = 300
```

2 Data Overview

This data set contains statistics of audio recordings of speech from both males and females. The data came from computer scientist Kory Becker's blog called Primary Objects. She gathered it

[1] https://en.wikipedia.org/wiki/Speech_recognition
[2] http://www.cs.columbia.edu/speech/PaperFiles/2016/levitan_prosody16.pdf
[3] http://asa.scitation.org/doi/abs/10.1121/1.401664, https://arxiv.org/abs/1003.4083
[4] https://en.wikipedia.org/wiki/Voice_frequency

from the following databases: Harvard-Haskins Database of Regularly-Timed Speech, Telecommunications & Signal Processing Laboratory (TSP) Speech Database at McGill University, VoxForge Speech Corpus, and Festvox CMU_ARCTIC Speech Database at Carnegie Mellon University. Ms. Becker preprocessed the data in R to get all the statistics for each sample[5]. She used machine learning on the data set to classify gender, but did no statistical analysis.

```
In [2]: voices = pd.read_csv("voice.csv")
```

The data set contains 3,168 instances and 20 features. Each instance represents the speaking voice of a person, and each feature is in kilohertz (kHz). The features themselves are statistics of different frequencies (fundamental and dominant) of the voice in the recording, such as mean, standard deviation, median, and first and third quartiles. Below is a summary of all the features. In the equations, N represents the dimensionality of the data, x_i a data point, and $\bar{x}$ the mean frequency of the sample.

1. `meanfreq`: Mean frequency of the voice.
2. `sd`: Standard deviation of the frequency.
3. `median`: Median frequency.
4. `Q25`: First quartile frequency.
5. `Q75`: Third quartile frequency.
6. `IQR`: Interquartile range frequency.
7. `skew`: Skew of the spectrum of the recording (measure of symmetry). It is calculated by the equation $\frac{N\sqrt{N-1}}{N-2}\frac{\sum_{i=1}^{N}(x_i-\bar{x})^3}{(\sum_{i=1}^{N}(x_i-\bar{x})^2)^{\frac{3}{2}}}$.
8. `kurt`: Kurtosis; the shape of the distribution of the voice sample. It is calculated by $\frac{N(N+1)(N-1)}{(N-2)(N-3)}\frac{\sum_{i=1}^{N}(x_i-\bar{x})^4}{(\sum_{i=1}^{N}(x_i-\bar{x})^2)^2}$.
9. `sp.ent`: Spectral entropy; the disorder/complexity of the voice sample. This measures the amount of disorder in a system. In other words, it measures the amount of large variations in the frequency spectrum. If $X(\omega_i)$ represents the spectrum of the signal, then $P(\omega_i) = \frac{1}{N}|X(\omega_i)|^2$ is the power spectral density of the signal. Let $p_i = \frac{P(\omega_i)}{\sum_i P(\omega_i)}$. Then the spectral entropy is $-\sum_{i=1}^{N} p_i \ln(p_i)$.
10. `sfm`: Spectral flatness (also called the tonality coeffecient or Wiener entropy). This measures the tonality of a sample. In other words, it measures the amount of peaks in a frequency spectrum. The less tonal a sample is, the more flat the spectrum will be (meaning it will resemble the flat spectrum of white noise). The smaller this value, the more tone-like the sound is. The higher the value (the closer to 1 it is), the more noise-like the sound. By "noise-like" we mean that the sample more closely resembles white noise (it doesn't have as high of an amplitude so the spectrum looks flat). Using the same variables that were used for spectral entropy, spectral flatness is calculated by $\frac{\exp(\frac{1}{N}\sum_{i=1}^{N}\ln(x_i))}{\bar{x}}$.
11. `mode`: Mode frequency.

[5] http://www.primaryobjects.com/2016/06/22/identifying-the-gender-of-a-voice-using-machine-learning

12. `centroid`: Centroid of the spectrum. Intuitively, this is the "brightness" of the audio (where brightness refers to the amount of high frequency signals present in the audio). This is the "center of mass" of the spectrum.
13. `meanfun`: Average of fundamental frequency (the lowest frequency) measured across the sample.
14. `minfun`: Minimum fundamental frequency measured across the sample.
15. `maxfun`: Maximum fundamental frequency measured across the sample.
16. `meandom`: Average of dominant frequency (the most-heard frequency, the frequency that carries the highest energy) measured across the sample.
17. `mindom`: Minimum of dominant frequency measured across the sample.
18. `maxdom`: Maximum of dominant frequency measured across the sample.
19. `dfrange`: Range of dominant frequency measured across the sample.
20. `modindx`: Modulation index (how much the modulated variable varies around its unmodulated level). Calculated as the accumulated absolute difference between adjacent measurements of fundamental frequencies divided by the frequency range.

An example of what the data looks like is below.

```
In [24]: voices.iloc[[0,1,2,-1,-2,-3]]
```

Out[24]:

	meanfreq	sd	median	Q25	Q75	IQR
0	0.059781	0.064241	0.032027	0.015071	0.090193	0.075122
1	0.066009	0.067310	0.040229	0.019414	0.092666	0.073252
2	0.077316	0.083829	0.036718	0.008701	0.131908	0.123207
3167	0.165509	0.092884	0.183044	0.070072	0.250827	0.180756
3166	0.143659	0.090628	0.184976	0.043508	0.219943	0.176435
3165	0.142056	0.095798	0.183731	0.033424	0.224360	0.190936

	skew	kurt	sp.ent	sfm	mode
0	0.370433	0.209530	0.893369	0.491918	0.000000
1	0.645731	0.484581	0.892193	0.513724	0.000000
2	0.885724	0.782619	0.846389	0.478905	0.000000
3167	0.049100	0.004405	0.938829	0.601529	0.267702
3166	0.045818	0.004114	0.950436	0.675470	0.212202
3165	0.054038	0.005043	0.946854	0.654196	0.008006

	centroid	meanfun	minfun	maxfun	ffrange	meandom
0	0.059781	0.084279	0.015702	0.275862	0.260160	0.007812
1	0.066009	0.107937	0.015826	0.250000	0.234174	0.009014
2	0.077316	0.098706	0.015656	0.271186	0.255531	0.007990
3167	0.165509	0.185607	0.062257	0.271186	0.208930	0.227022
3166	0.143659	0.172375	0.034483	0.250000	0.215517	0.791360
3165	0.142056	0.209918	0.039506	0.275862	0.236356	0.494271

	mindom	maxdom	dfrange	modindx	label
0	0.007812	0.007812	0.000000	0.000000	male
1	0.007812	0.054688	0.046875	0.052632	male
2	0.007812	0.015625	0.007812	0.046512	male
3167	0.007812	0.554688	0.546875	0.350000	female
3166	0.007812	3.593750	3.585938	0.311002	female
3165	0.007812	2.937500	2.929688	0.194759	female

3 Data Cleaning

The data has no missing values (see code below) or major outliers (see the data visualization section).

```
In [5]: voices.isnull().values.any()
```

```
Out[5]: False
```

We check the number of female instances versus male instances. If one gender has significantly more instances than the other, the statistics for one gender could be more accurate than the statistics for the other. We find that there are equal numbers of male and female instances, so we do not have to delete, duplicate, or randomly sample instances to make them equal.

```
In [6]: Counter(voices['label'])
```

```
Out[6]: Counter({'female': 1584, 'male': 1584})
```

The data seems to be unbiased, especially since the samples come from multiple sources. We check the data to ensure all observations are positive, since a negative frequency does not make sense in the context of human speech (see the code below). We also check that the data are within the correct ranges (e.g. the spectral flatness values are all between 0 and 1), as seen below in the minimum and maximum values for each column.

```
In [37]: print("Feature\t\tMin\t\tMax")
         for col, val1, val2 in zip(voices.columns, voices.min(), voices.max()):
             print(str(col) + '\t\t' + str(val1) + '\t' + str(val2))
```

```
Feature                 Min                 Max
meanfreq                0.0393633425836     0.25112375872
sd                      0.0183632424445     0.115273246744
median                  0.0109745762712     0.261224489796
Q25                     0.000228758169935   0.247346938776
Q75                     0.0429462738302     0.273469387755
IQR                     0.0145577312627     0.252225201072
skew                    0.00408160040571    1.0
kurt                    0.00157944039097    1.0
sp.ent                  0.738650686224      0.981996588964
sfm                     0.0368764745063     0.842935931447
mode                    0.0                 0.28
```

```
centroid                0.0393633425836    0.25112375872
meanfun                 0.0555653493135    0.237636387269
minfun                  0.00977517106549   0.204081632653
maxfun                  0.103092783505     0.279113924051
ffrange                 0.0275807722616    0.268002606712
meandom                 0.0078125          2.95768229167
mindom                  0.0048828125       0.458984375
maxdom                  0.0078125          21.8671875
dfrange                 0.0                21.84375
modindx                 0.0                0.932374100719
```

To use this data set in machine learning applications, we change the labels to numerical values for later convenience. Originally these labels were "male" or "female", which we change to 0 and 1, respectively.

```
In [7]: voices.label = voices.label.map(dict(female = 1, male = 0))
```

Many of the values came pre-normalized. The only columns that are not normalized are the dominant frequency, skew, and kurtosis. We do not normalize the dominant frequency data as we do not have the actual dominant frequency values, only the statistics associated with them. But we do normalize the skew and the kurtosis.

```
In [4]: voices['kurt'] = voices['kurt']/voices['kurt'].max()
        voices['skew'] = voices['skew']/voices['skew'].max()
```

The `centroid` and `meanfreq` columns are identical. As it does not provide additional insight, we drop the `centroid` column.

```
In [8]: print("Proportion of equal values: {}".format(len(np.where(voices['centroid
        voices = voices.drop('centroid', axis=1)

Proportion of equal values: 1.0
```

4 Feature Engineering

As exhibited by the analysis above, these features cover a wide variety of measurements for an audio signal. Since we do not have access to the original signal, there are not many features we can engineer. The only feature we do engineer is the `ffrange` feature. Since the original feature set includes the range of dominant frequency measured across the sample (`dfrange`), we add the feature `ffrange` to represent the range of fundament frequency across the sample. To do this, we subtract the minimum fundamental frequency from the maximum fundamental frequency.

```
In [9]: voices.insert(15, 'ffrange', voices['maxfun'] - voices['minfun'])
```

These features are sufficient to analyze the audio signals and to classify gender using machine learning algorithms.

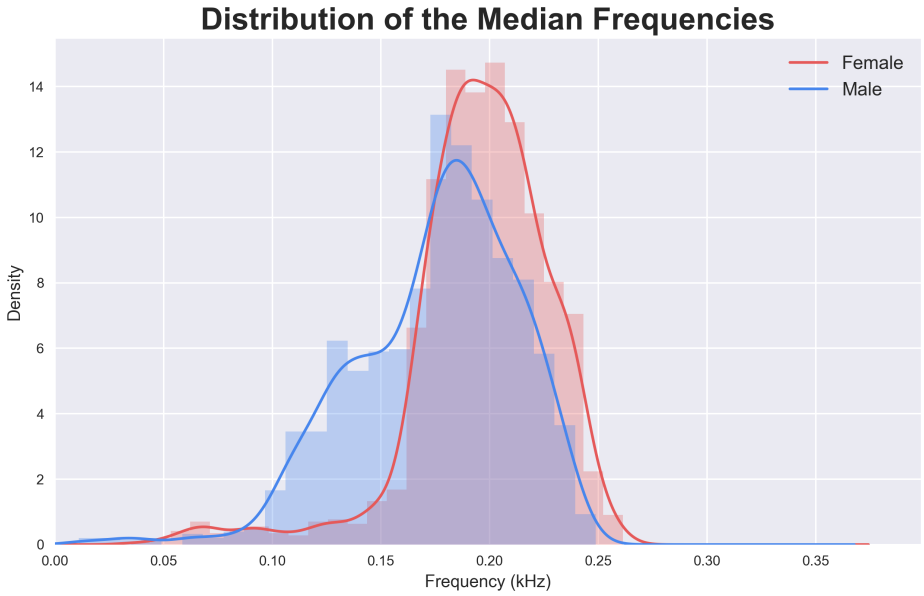
5 Data Visualization

In this section, we look for trends in the data using different visualizations. We start by looking at the statistics for female and male speaking frequencies. We are interested to see if there is a tangible difference between them, outside the range of fundamental frequency.

```
In [11]: # rot determines the rotation, fs the fontsize.
         rot, fse = 0, 9
         red, blue = '#e55959', '#4786ed'
         females = voices[voices['label'] == 1]
         males = voices[voices['label'] == 0]
         def plot_distribution(col, hist=True, title=None, bins=25):
             """Plot the distribution for a given attribute.
             Inputs:
                 col (str): Attribute to plot.
                 hist (bool): Whether histograms should be plotted.
                              Defaults to True.
                 title (str): Title of the graph. Defaults to None.
                 bins (int): Number of bins to use in histograms."""
             if hist:
                 females[col].plot(kind='hist', bins=bins, alpha=0.3,
                                   rot=rot, color=red,
                                   fontsize=fs, normed=True, label='')
                 males[col].plot(kind='hist', bins=bins, alpha=0.3,
                                 rot=rot, color=blue,
                                 fontsize=fs, normed=True, label='')
             females[col].plot(kind='kde', color=red, rot=rot, fontsize=fs,
                               label='Female')
             males[col].plot(kind='kde', color=blue, fontsize=fs,
                             rot=rot, label='Male')
             plt.xlabel("Frequency (kHz)")
             plt.ylabel("Density")
             if not title:
                 title = col
             plt.title("Distribution of the {}".format(title), fontsize = 20,
                       fontweight = 'bold')
             plt.legend(loc = 'upper right', prop = {'size': 12})
             plt.gca().set_xlim(left=0)
```

To confirm that, in general, female voices have higher fundamental frequencies than males, we plot the median frequencies.

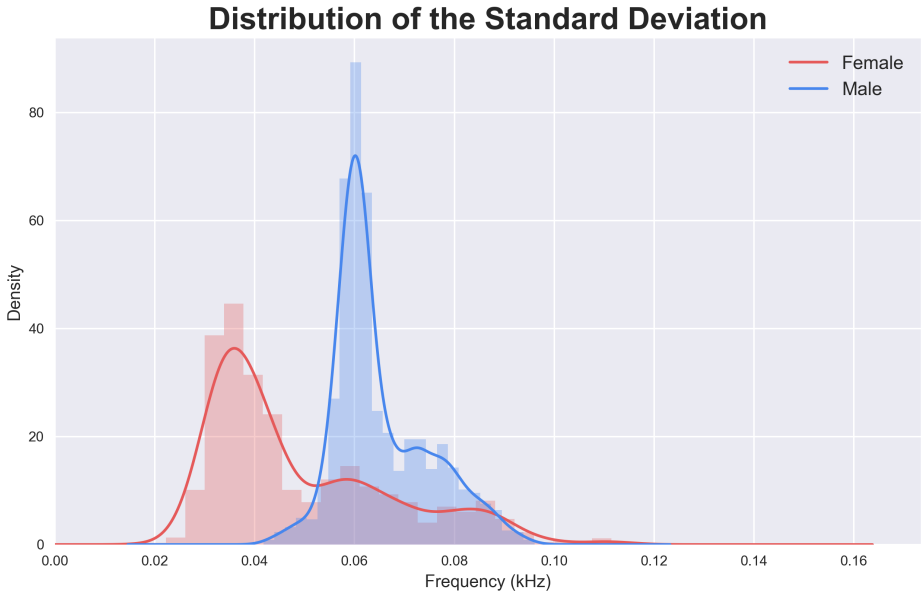
```
In [12]: plot_distribution('median', True, "Median Frequencies")
```



The graph above suggests that, in general, median female fundamental frequencies are indeed higher than male frequencies. However, their distributions contain a significant overlap, especially since the data is normalized. Therefore, fundamental frequency cannot be the only factor used to determine the gender of a voice sample.

Additionally, it seems that the median frequencies for females are more centrally distributed than for males. To investigate this hypothesis, we examine the standard deviation for each gender.

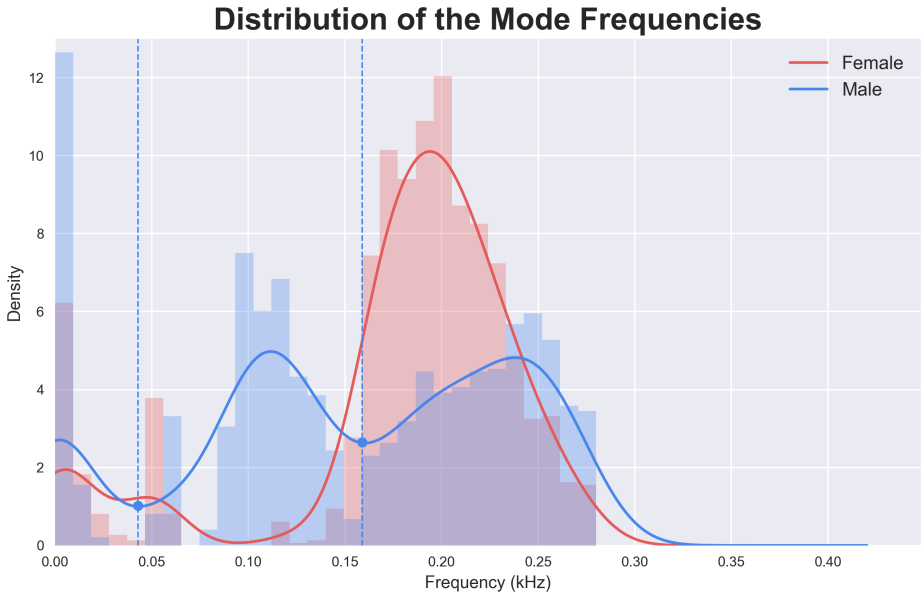
```
In [13]: plot_distribution('sd', True, "Standard Deviation")
```



This visualization supports the hypothesis that female frequencies are more centrally distributed than male frequencies. In fact, a large amount of the male samples have standard deviations around 0.060, while most female samples have standard deviations of 0.036. According to these findings, one could suggest that males tend to vary their speech more as they talk.

One of the most insightful discoveries about this data came from analyzing the mode frequencies.

```
In [16]: plot_distribution('mode', True, "Mode Frequencies", bins=30)
         # Plot the vertical lines where the splits occur.
         plt.axvline(x=0.043, ls='dashed', lw=1, color=blue)
         plt.axvline(x=0.159, ls='dashed', lw=1, color=blue)
         # Plot the intersections of the KDEs and the splits.
         plt.plot([0.043], [1.02], marker='.', color=blue, ms=13)
         plt.plot([0.159], [2.65], marker='.', color=blue, ms=13)
```

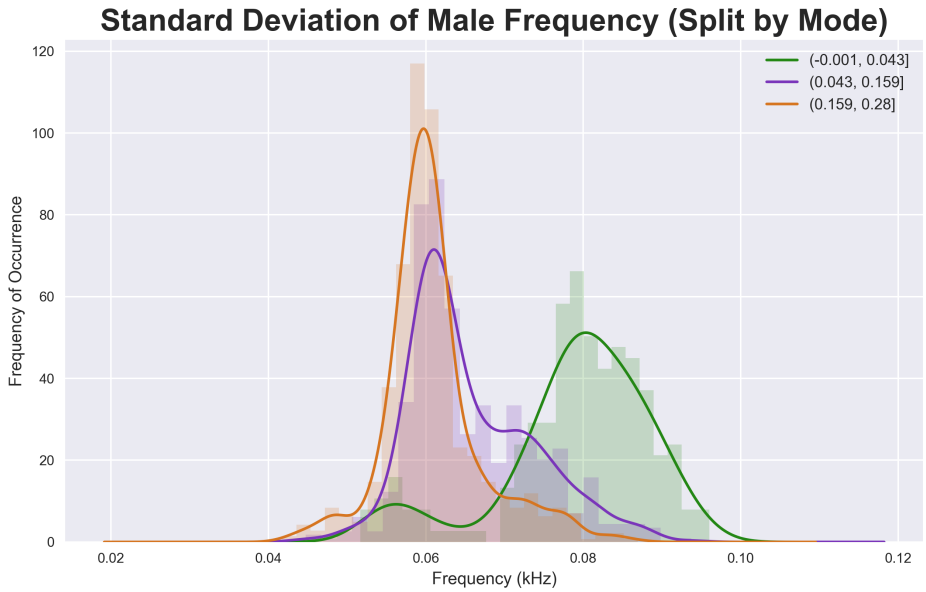


This graph splits the males into three different categories. In the bottom category, the majority of the modes are zero. This suggests that there exists a significant amount of silence in these sound samples (we assume that a value around zero implies silence). If we had access to the original data, we would check to make sure these instances have enough information to be interpreted. But since we do not, and we also do not know how the data points are binned (if the bins are small, a small amount of silence could be the mode of distribution), we leave these observations in the data and analyze them as a group.

We now examine the differences between these three groups of males. Because the female mode frequencies were essentially split into two groups (nearly silence and normal speaking frequency), we omit an analysis of females split by mode frequencies (the results are not insightful). We first compare the standard deviations of these three groups.

```
In [21]: # Create a new column for males with their grouping by mode.
         males['split'] = pd.cut(males['mode'], [males['mode'].min(), .043,
                                 .159, males['mode'].max()], include_lowest = True)
         green, purple, orange = "#238714", "#7a35ba", "#d67520"
         colors = np.array([green, purple, orange])
         for i, group in enumerate(males.groupby('split')):
             group[1]['sd'].plot(kind='hist', bins=25, alpha=0.2, rot=rot,
                                 color=colors[i], fontsize=fs, normed=True, label='')
```

```
    group[1]['sd'].plot(kind='kde', color=colors[i], label=group[0])
plt.ylabel("Frequency of Occurrence")
plt.xlabel("Frequency (kHz)")
plt.legend(loc='upper right')
plt.title("Standard Deviation of Male Frequency (Split by Mode)",
          fontsize=20, fontweight='bold')
```

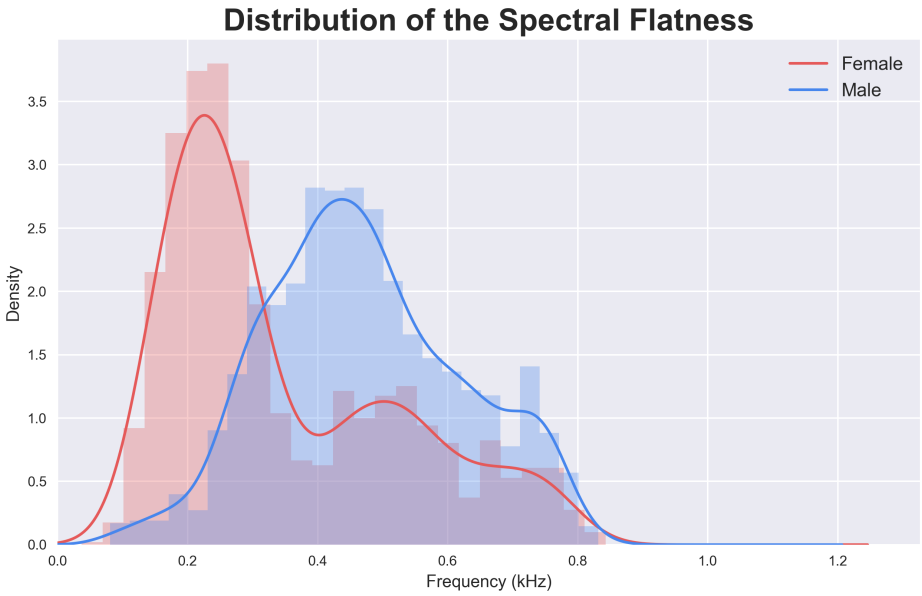


We refer to the lowest-valued mode group as Group A, the middle-valued mode group as Group B, and the highest mode group as Group C. The highest standard deviation of all three groups occurs in Group A. This makes intuitive sense, as zeros are far away from the mean frequency; even though zero is the mode frequency, there must be enough nonzero frequencies to pull up the standard deviation. Note that the minimum standard deviation of Group A is higher than the minimum standard deviations of the other two groups, implying that the majority of the samples are not just silence.

The standard deviation graph for these three groups is the most interpretable graph given all features. The other graphs are omitted as they do not add to this analysis.

We now compare the spectral flatness between males and females.

```
In [22]: plot_distribution('sfm', True, "Spectral Flatness")
```

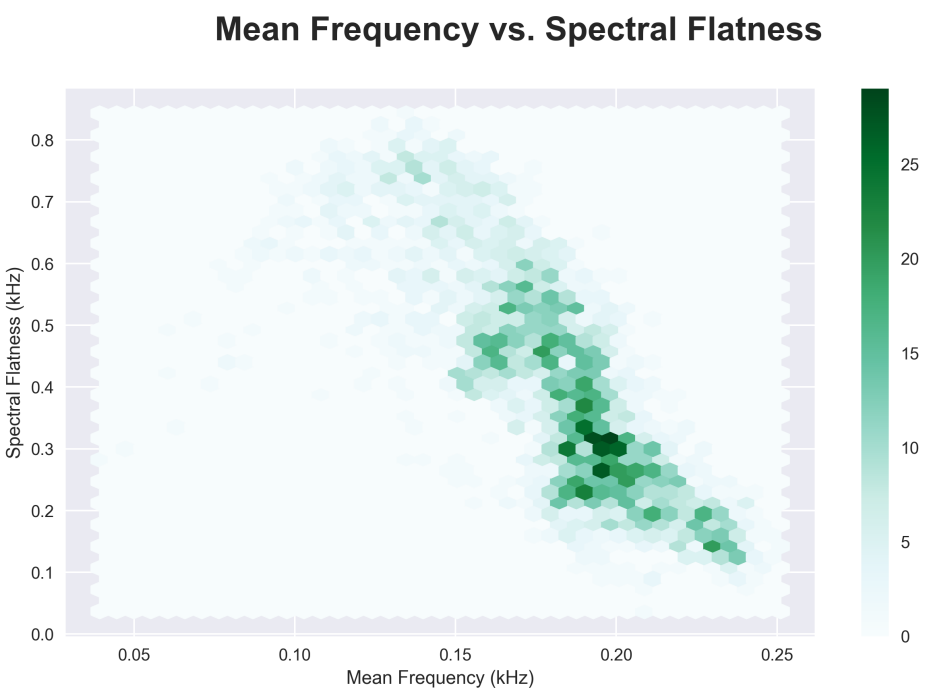


For females, the distribution skews to the left, while the male distribution is more normal. Recall that as the spectral flatness approaches one, the voice is more "noise-like". The closer the spectral flatness is to zero, the more tone-like the voice. This insinuates that female voices tend to be more tonal, and that male voices are more evenly spread from tonal to noisy values for the spectral flatness. This means that females' voice waves have much higher peaks than males. As the male voice waves have shorter peaks, they more closely resemble white noise.

So far, we have discovered that there are differences between the male and female distributions for the median, standard deviation, mode, and spectral flatness of a voice sample. Consequently, all of these features could be used to perform gender recognition from voice samples.

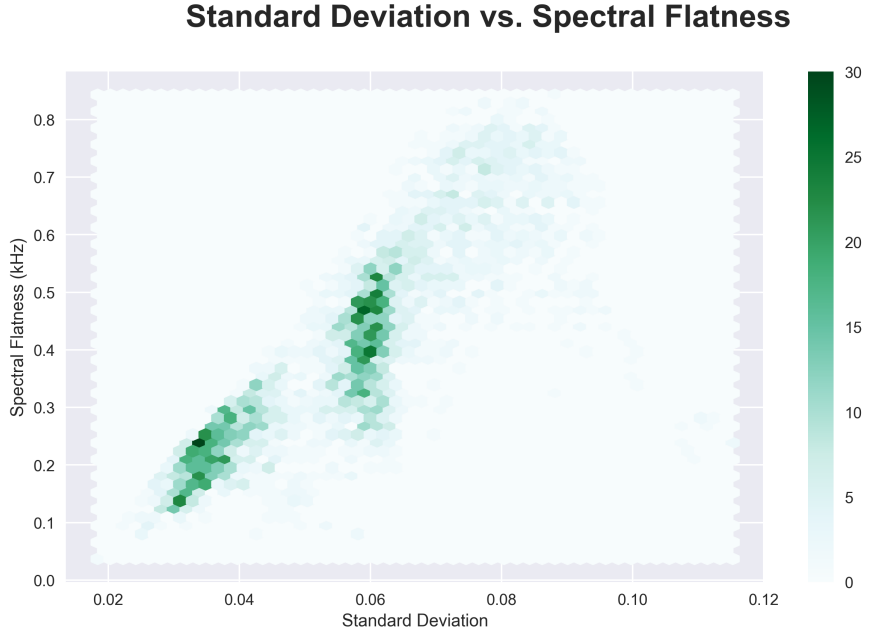
We now study how various acoustic statistics relate to one another, to answer the second question. We include the relationships where a correlation exists and is non-trivial.

```
In [24]: voices.plot(kind= "Hexbin", x="meanfreq", y="sfm", gridsize=40)
         plt.suptitle("Mean Frequency vs. Spectral Flatness", fontsize=20,
                      fontweight='bold')
         plt.xlabel('Mean Frequency (kHz)')
         plt.ylabel('Spectral Flatness (kHz)')
```



The majority of the voices have higher frequencies and lower spectral flatness values. In other words, most voices sound more tone-like than noise-like, especially when they talk fast. This makes intuitive sense, as more tone-like waves means that the waves have higher amplitudes. We now compare the tone-like quality of a voice to standard deviation.

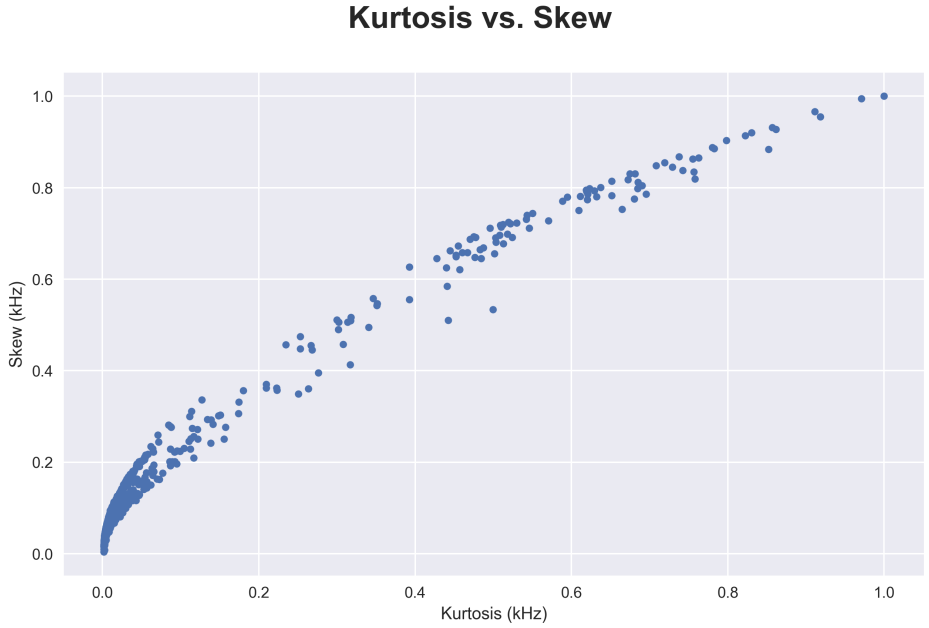
```
In [25]: voices.plot(kind = "Hexbin", x = "sd", y = "sfm", gridsize = 50)
         plt.suptitle("Standard Deviation vs. Spectral Flatness", fontsize=20,
                      fontweight='bold')
         plt.xlabel('Standard Deviation')
         plt.ylabel('Spectral Flatness (kHz)')
```



This graph shows two main clusters of values. One has a low spectral flatness and standard deviation, whereas the other has mid-range values. This graph also suggests a slight positive correlation between spectral flatness and standard deviation. As the standard deviation increases, the spectrum comes closer to resembling that of white noise (a flat spectrum). This does not make intuitive sense as one would assume that increasing the standard deviation of a sample would increase the variations in frequency.

Finally, we examine the relationship between skew and kurtosis, whose equations are similar.

```
In [29]: voices.plot(kind="scatter", x="kurt", y="skew")
         plt.xlabel('Kurtosis (kHz)')
         plt.ylabel('Skew (kHz)')
         plt.suptitle("Kurtosis vs. Skew", fontsize=20, fontweight='bold')
```



There seems to be an almost log-like correlation between the skew and kurtosis. As the shape of the distribution increases, the skew does as well.

6 Conclusion

There does seem to be differences between some statistics of these voice samples based on gender, most notably in the distributions of the median, standard deviation, mode frequencies, and the spectral flatness. These results show that female voices tend to have higher frequencies and their frequency distributions are more tone-like. These differences could be significant enough for a machine learning algorithm to classify whether or not a voice is male or female (which will be explored in future work).

We also found that various statistics on audio recordings relate to each other. We found a negative correlation between spectral flatness and mean frequency. On the other hand, we found a negative correlation between spectral flatness and standard deviation. Lastly, we found an almost log-like relationship between kurtosis and skew, a relationship not intuitive from their equations.